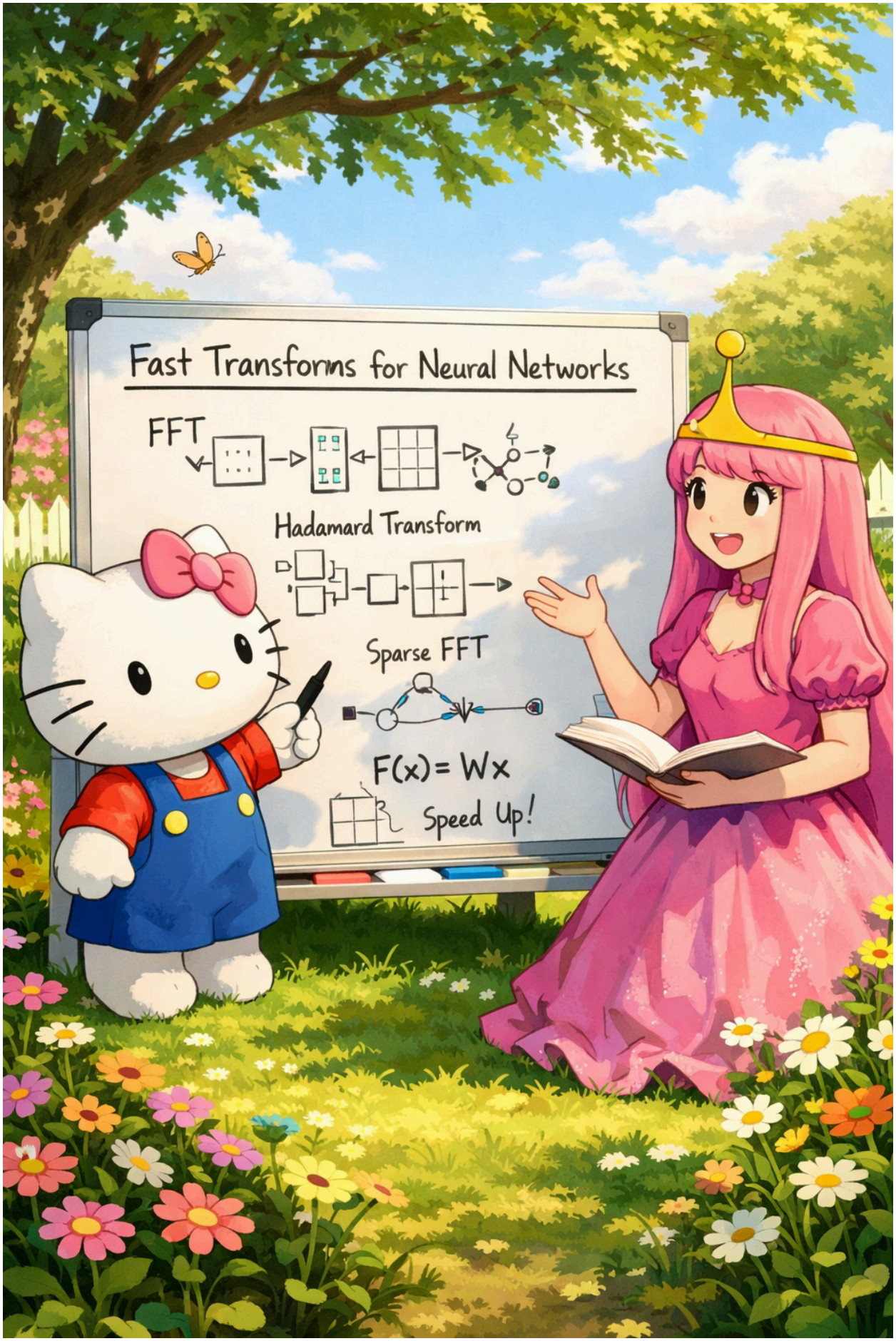
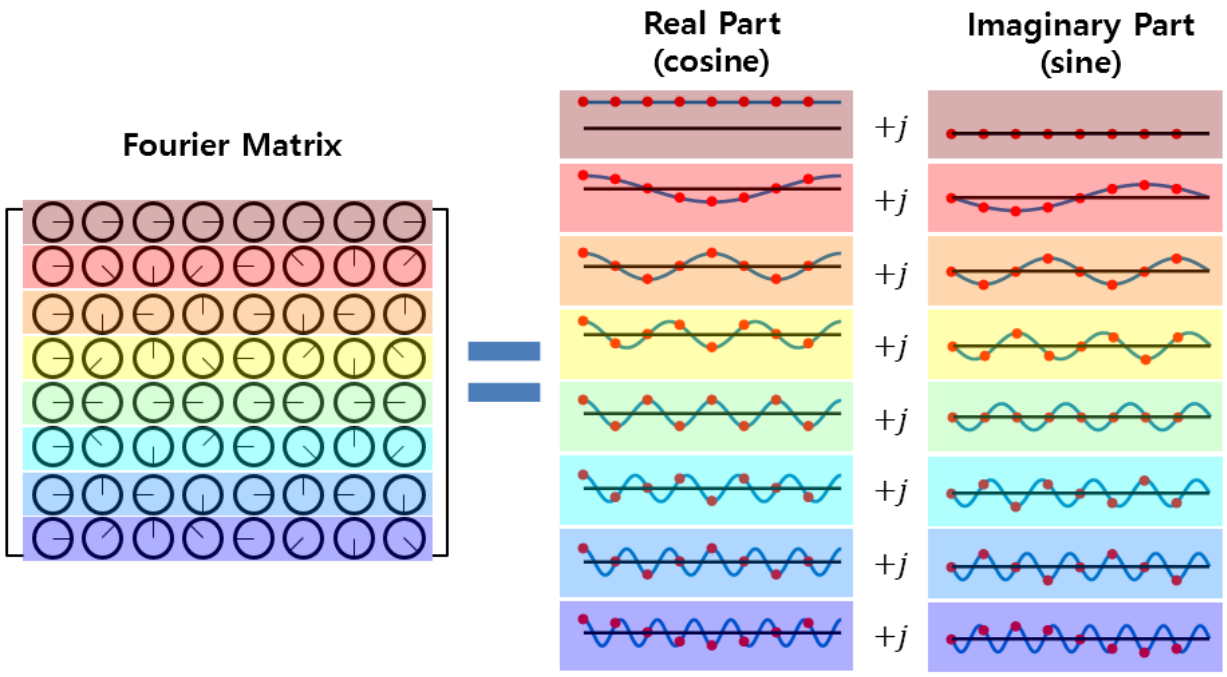
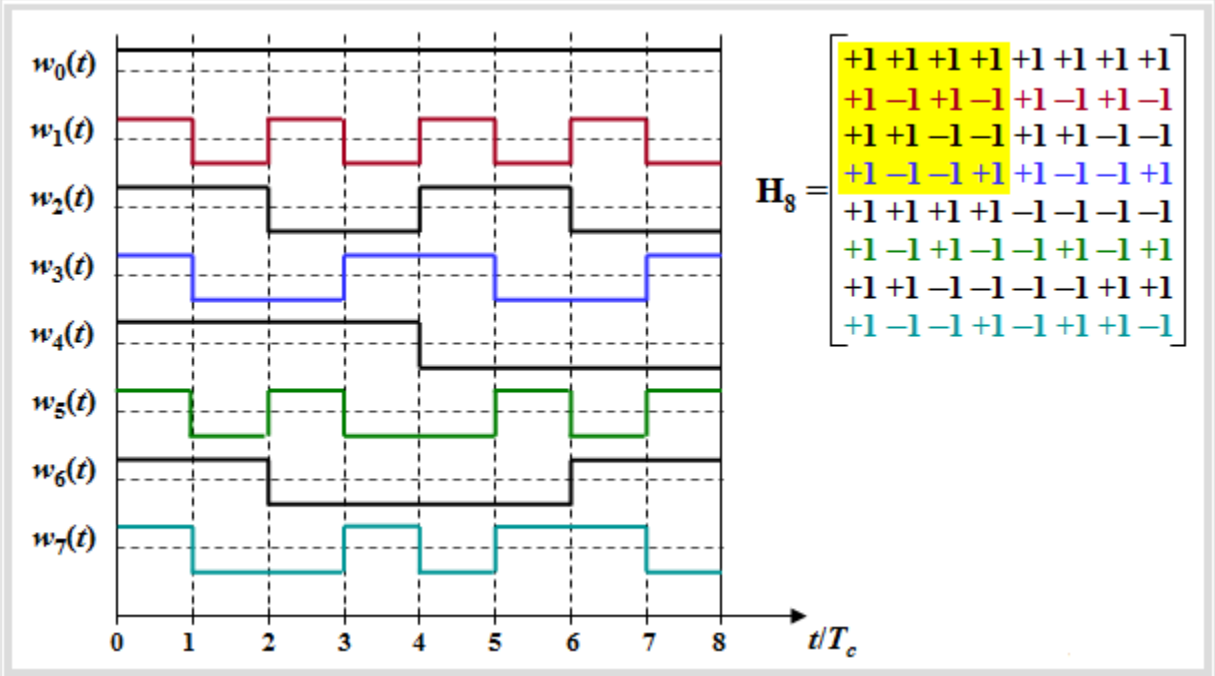

Fast Transforms for Neural Networks
FFT
Hadamard Transform
Sparse FFT
F(x) = Wx
Speed Up!

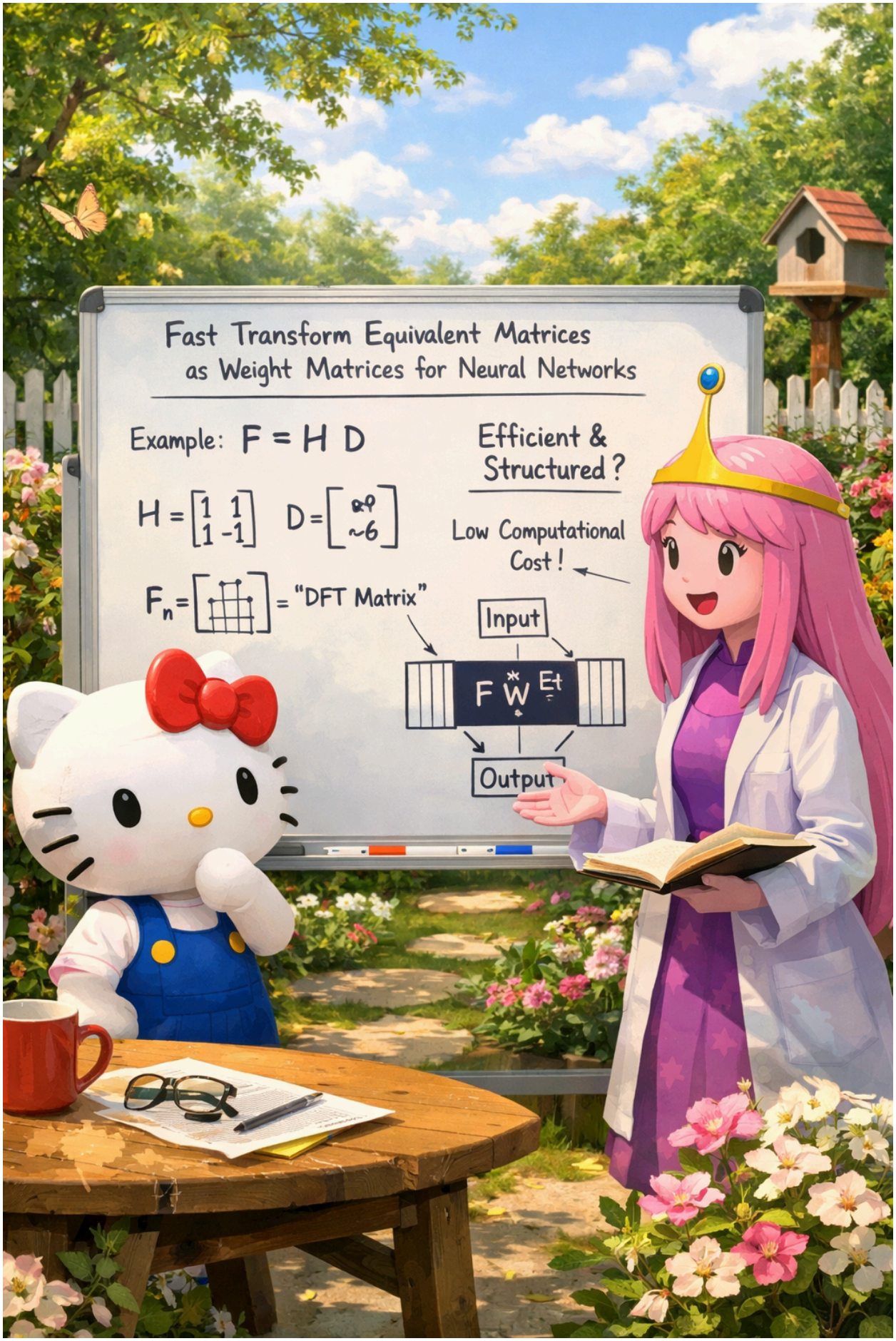
Real Sine Matrix
Real Cosine Matrix
Walsh-Hadamard Matrix
?



Imaginary part can be viewed & computed as real.



Walsh Hadamard matrix H8.


Fast Transform Equivalent Matrices
as Weight Matrices for Neural Networks
Example: F = H D
H = [1 1; 1 -1]
D = [α 0; 0 ~6]
Efficient & Structured?
Low Computational Cost!
F_n = [ ] = "DFT Matrix"
Input
F W Et
Output

Fast Transform (matrix equivalent) as neural network matrix:
Advantages: nlog2(n) compute cost vs n^2, fully connected architecture.
Disadvantage: Frozen neural network.
Fixes:

1. Parametric (adjustable) activation functions.
2. Adjustable diagonal matrix prefix.
3. Use multiple small width layers as vector to vector parametric functions.

Technicalities:
1 & 2 require width to be increased by 4, 8 or 16 to allow proper information flow.
Item 3 does not require any widening.
Spectral bias of the fast transforms at the input and output of the neural network can be dealt with by using random or sub-random sign flips. Eg. Permuted Thue-Morse sequence. No spectral issues internal to the neural network.

Avoiding Spectral Bias in Fast Transforms
• Solution: Random Sign Flipping!
??
• Random or Sub-Random ±1 Multipliers
+1/-1/+1/-1...
X * ε -> ±X
Input X
±1, ±1, ±1
Transform
Output F(X)

Good Luck
With Your
Research!
E=mc²